ELECTRONICS LABORATORY • SYRACUSE, N.Y.



# MODIFICATIONS TO INTERIM VISUAL SPACEFLIGHT SIMULATOR

FINAL REPORT

22 February 1968

CONTRACT NO. NAS 9-3916

Prepared for

NASA MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

GENERAL ELECTRIC

MODIFICATIONS TO INTERIM VISUAL SPACEFLIGHT SIMULATOR

Contract No. NAS 9-3916

Final Report

22 February 1968

Prepared for

NASA Manned Spacecraft Center
Houston, Texas

Prepared by

Electronics Laboratory
Defense Electronics Division
GENERAL ELECTRIC COMPANY
Syracuse, New York

## ABSTRACT

This is the Final Report submitted in accordance with the provisions of Contract NAS 9-3916, Modifications to Interim Visual Spaceflight Simulator.

The program encompasses the design, fabrication, and installation of additions and modifications to the Interim Visual Spaceflight Simulator located at the NASA Manned Spacecraft Center, Houston, Texas. The additions and modifications comprise:

1) Addition of a three-dimensional object generating capability;
2) Addition of the capability for independent operation of the three displays;
3) Increased brightness, improved convergence, and higher resolution;
4) Addition of a rendezvous vehicle beacon; and
5) Addition of input-output equipment.

This report contains a summary of the Contract effort and a description of the delivered equipment and the mathematics involved in its operation.

# TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

# 1.0 SYSTEM DESCRIPTION

## 1.1 TASK DESCRIPTION

The Electronic Scene Generator is an electronic system which simulates the external visual environment of a space vehicle. Effort on Contract NAS 9-3916 has been directed toward producing equipment which will operate in conjunction with that furnished under Contract NAS 9-1375 to provide for the generation of three-dimensional objects, production of certain special effects, improvement of display quality, and expansion of input interface capability. Special pictorial effects provided include horizon dip and curvature, a moving vehicle's shadow, and a point-source flashing beacon. The expanded interface capability allows communication with up to five source computers furnishing attitude and position data for multivehicle studies or for several studies run simultaneously.

## 1.2 SYSTEM FUNCTIONAL DESCRIPTION

Digital descriptions of the location, size, shape, and color of objects used to simulate the visual environment are stored in the Electronic Scene Generator (ESG). Inputs to the equipment describe the attitude and position of from one to three observers in the environment, and perspective transformations are performed at the frame rate of the kinescope displays to present each observer with his correct perspective view of the simulated environment.

A block diagram of the System is shown in Figure 1. This diagram shows the principal functional units of the ESG and indicates the relationship of new to existing equipment. Figure 2 shows single-bay cabinet front and rear views.

Input-output functions for environment description and program data are performed by the Raytheon 520. This computer also performs some coordinate rotation computations.

The Input Interface Unit (IIU) handles all signals between source computers and the ESG. The IIU interrogates source computers specified by the R520, formats the data, and stores the data words for transfer to the R520.

The Digital-to-Analog Conversion Unit (DACU) converts to analog form selected quantities from ESG memory for transmission to readout devices, such as panel meters, or to source computers. Quantities to be converted are automatically scaled by the R520 to make efficient use of available dynamic range. Twelve channels of conversion are available.

The Vector Calculating Unit (VCU) is a special purpose, high-speed, digital computer specifically designed to perform certain vector calculations. Arithmetic operations on the vector components are performed by three parallel units. Four core memory units are used; one for program storage

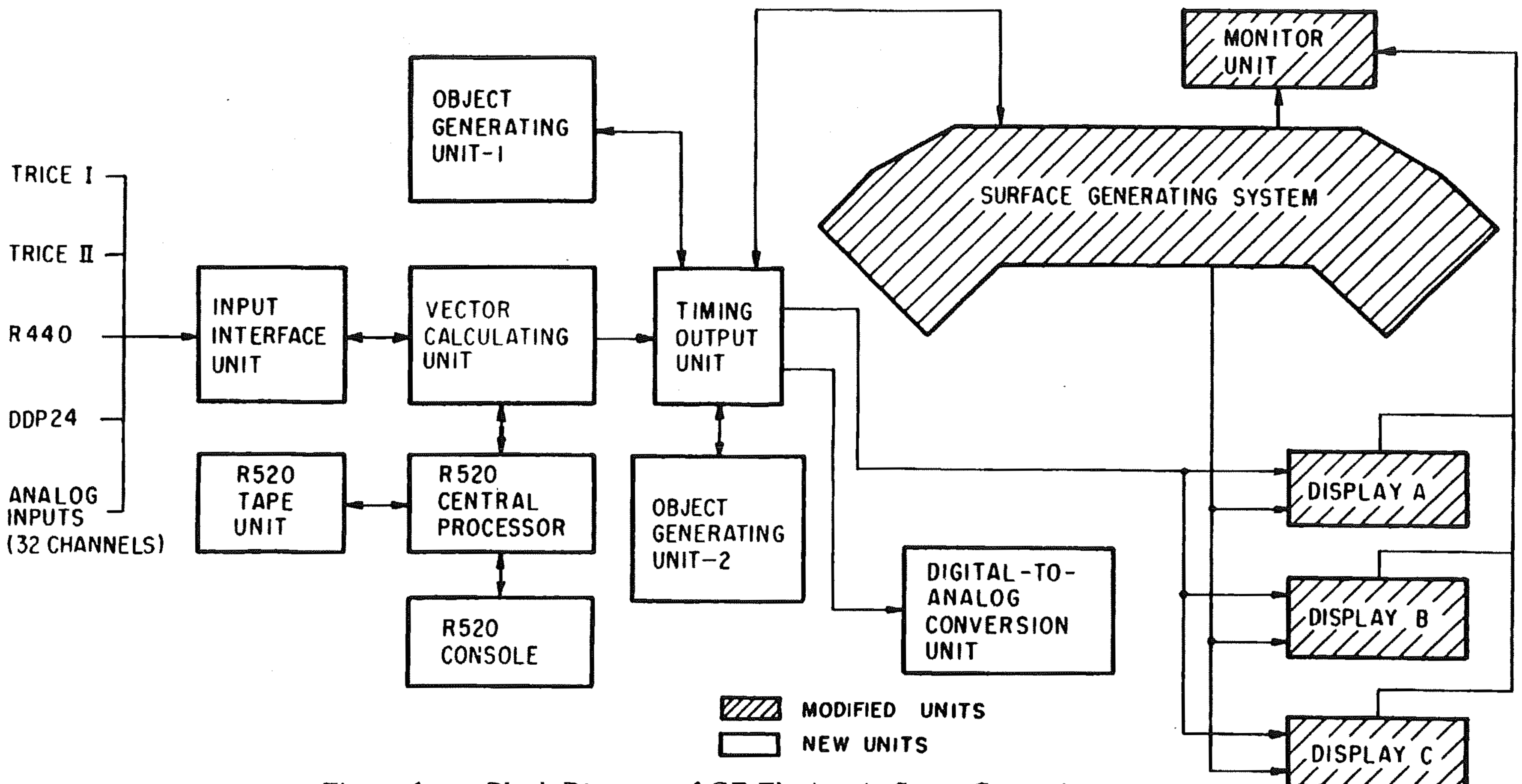


Figure 1. Block Diagram of GE Electronic Scene Generator

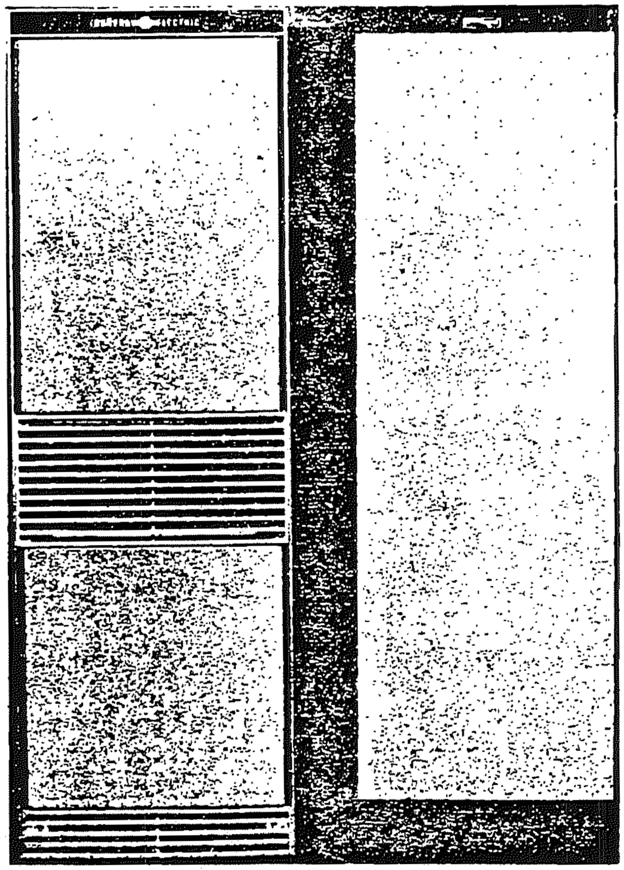

Figure 2. Typical View of a Cabinet

and three for data storage. The VCU also serves as an input-output junction for data transmitted between the Arithmetic and Logic Subsystem and other portions of the ESG.

The VCU's primary function is to determine the display-plane image of the object environment. Data describing the objects are stored in the arithmetic memory. Object vertex data are used to define the edges which bound the faces of the objects, and the image of each edge is computed in the appropriate display plane coordinates. The perspective (i. e., front or back view) the observer has on each plane in the environment is determined. Finally, the position of each object relative to the station point is compared with the position of all other objects to determine which objects will obscure which other objects.

Hardware of the Object Generating Subsystem (OGS) is divided into the identical units--OGU-1 and OGU-2. Figure 3 shows OGU construction and wiring. Each OGU is capable of working with up to 20 objects consisting of up to 80 faces and described by 120 edges. The function of an OGU is to scan the display plane image of the objects in synchronism with the display raster and to provide digital outputs designating the color of each element on the display as the scene is scanned out.

The Timing and Output Unit (TOU) has three major functions. It generates the basic timing for the ESG, serves as a mixing and distribution point for the video outputs from the two OGU's, and generates the test pattern and timing signals used for system alignment and trouble-shooting.

Modifications to the Surface Generating Subsystem (SGS) have been implemented to allow its operation at 20 frames per second, to synchronize its timing with that generated in the TOU, and to bring its operation under control of the R520. The SGS has been relieved of much of its previous computing load, particularly the problem-dependent coordinate rotations, and therefore executes a largely fixed program under control of the R520.

Displays and monitors in the Display Electronics Subsystem (DES) have been modified to allow 20 frames-per-second operation and to improve picture quality. To accommodate object video and to improve signal rise time, the Video Processor Unit (VPU) and Video Amplifier have been replaced by improved units. The new VPU also places all color value selection under program control. The Convergence Generator in the displays has also been replaced by an improved unit. Buffer amplifiers in the monitors have been modified to improve their rise time to accommodate increased object bandwidth.

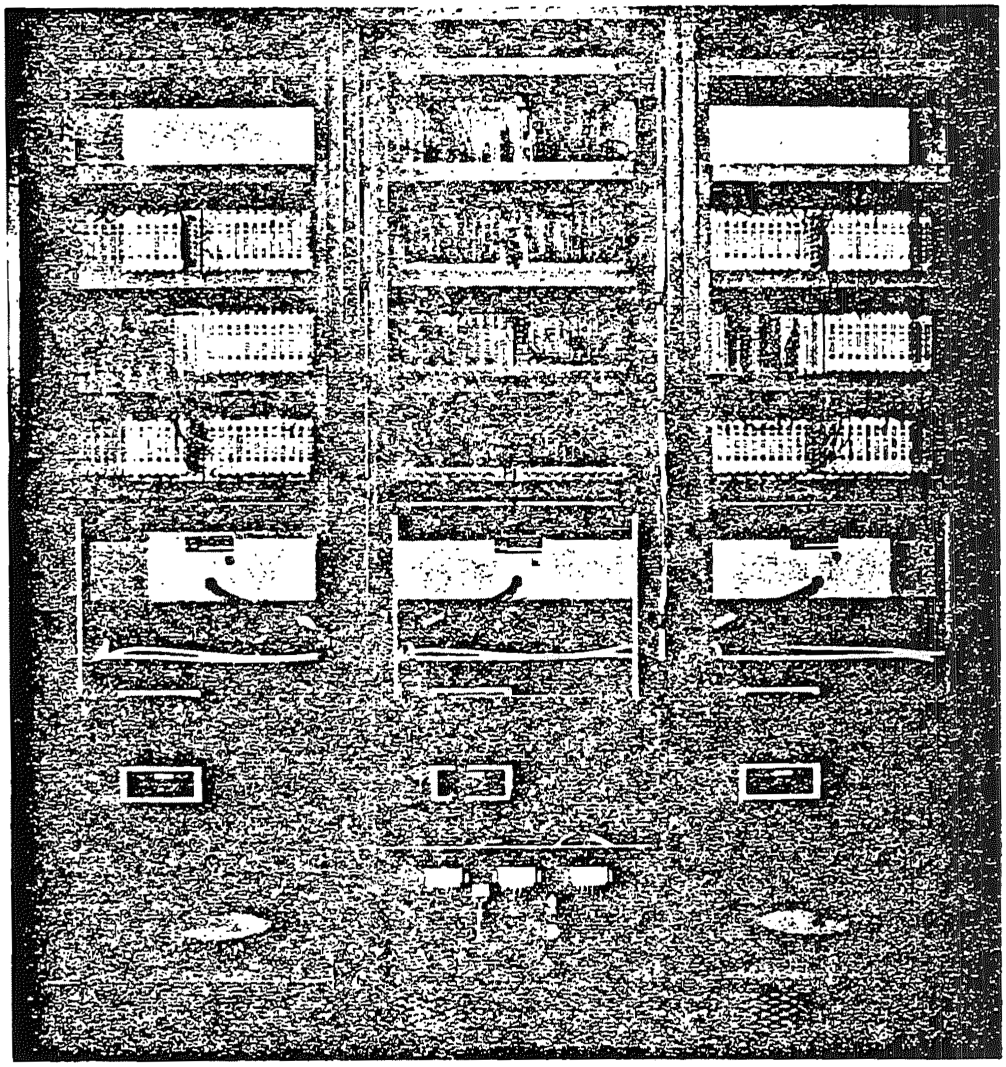

Figure 3. View of Typical Cabinet Wiring

## 2.0 MATHEMATICAL DESCRIPTION

The central mathematical task is to compute the image of each edge in the environment. The bulk of VCU computations are concerned with this task although the VCU also provides perspective and priority information to the OGS. The R520 computations are primarily coordinate rotations devoted to obtaining the basic vector sets needed for edge calculations. The following discussion makes use of a system of notation which is described in Section 1 of the Instruction Manual for Modifications to Interim Visual Space-Flight Simulator, Volume I.

### 2.1 VCU ARITHMETIC OPERATIONS

#### 2.1.1 Image Formation

The objects in the environment are convex polyhedra. They have plane faces bounded by straight edges. Consider first the problem of finding the display plane image of a single face.

Suppose that the vertices defining the face are numbered 1-2-3-4 as shown in Figure 4. In order to draw the image of this face, we need to know when the ray defined by the P Vector from the station point, 0, to the scanning spot, P, passes through the plane face 1-2-3-4. Equivalently, we need to know the ray is simultaneously below edge 1-2, to the left of edge 2-3, above edge 3-4, and to the right of edge 4-1. Note that these edges are considered as infinite lines, determined by their vertex pairs.

Consider the edge 1-2 and its image. The vectors from the station point to the vertices are VL vectors. The scanning spot is below edge 1-2 when and only when the scalar triple product $P \cdot (VL1 \times VL2) > 0$. Call this condition $F_1$. Similarly,

$$\left.\begin{array}{l} \text{P left of 2-3 is equivalent to } P \cdot (VL2 \times VL3) > 0 \text{ and is called } F_2 \\ \text{P above 3-4 is equivalent to } P \cdot (VL3 \times VL4) > 0 \text{ and is called } F_3 \\ \text{P right of 4-1 is equivalent to } P \cdot (VL4 \times VL1) > 0 \text{ and is called } F_4 \end{array}\right\} \quad (1)$$

Therefore, the condition that the P vector be inside of the image of face 1-2-3-4 is $F = F_1F_2F_3F_4$, where juxtaposition denotes logical AND.

If the side of the face where the vertices appear in clockwise order, as above, is called the front face, then F is the condition for drawing the front face. Call 4-3-2-1 the back face, and say that B is the condition that the extended P vector pierces the face through the back side. Then $B = \bar{F}_1\bar{F}_2\bar{F}_3\bar{F}_4$, where the bar denotes logical negation. By convention we will say that the back side of a face is that side interior to a polyhedron; for plane polygons,

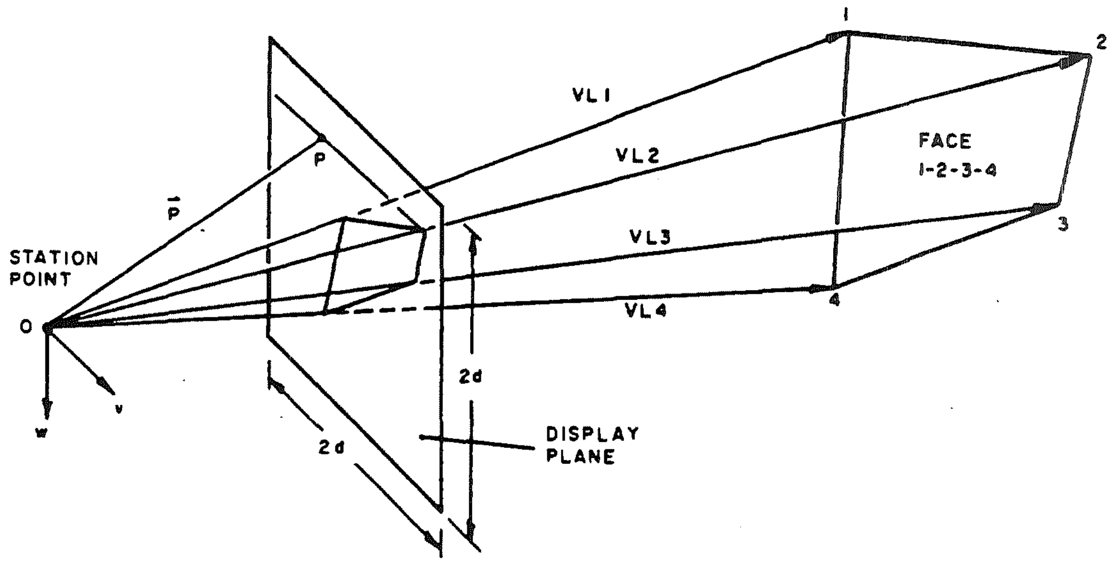


Figure 4. Image Formation

it is the face side whose vertices are numbered counter-clockwise, as in this example. Back faces of polyhedra are seen only from within the polyhedron or through a face designated as transparent. (One face of each polyhedron may be made transparent by operating that object in the transparent mode or the crater mode.)

For a given edge, say 1-2, we must monitor the sign of $Q = P \cdot (VL1 \times VL2)$. $Q < 0$ indicates that the scanning ray is on one side of the edge; $Q > 0$ indicates that the scanning ray is on the other side. Alternatively, we may look for $Q = 0$, knowing the direction (positive or negative) in which Q is changing. The scanning spot, defined by the P vector, follows a raster pattern moving across successive elements on each consecutive line. Q may be expressed in terms of the raster parameters by stating the P vector in terms of its starting location, PC, and its line and element dependent components PL and PE respectively. Expressing P in terms of line and element numbers:

$$P = PC + IPL - JPE \tag{2}$$

Then:

$$Q = P \cdot (VL1 \times VL2) = PC \cdot (VL1 \times VL2) + IPL \cdot (VL1 \times VL2) - JPE \cdot (VL1 \times VL2) \tag{3}$$

or

$$Q = QC + IQL - JQE \tag{4}$$

where

$$QC = PC \cdot (VL1 \times VL2)$$

$$QL = PL \cdot (VL1 \times VL2)$$

$$QE = PE \cdot (VL1 \times VL2)$$

Suppose that $QE = 0$. Then $Q = QC + IQL$ changes at most each line period. Its sign can be checked at the beginning of each raster line. This situation arises when an edge image is exactly parallel to a raster line.

Suppose that $QE \neq 0$. Then $Q = QC + I\,QL - J\,QE = 0$ when

$$J = J_0 = \frac{QC}{QE} + I\,\frac{QL}{QE} \quad .$$

$J_0$ is the element number for the particular line I at which the scanning ray intersects the edge. The direction in which Q is changing as a scan line is traversed is given by the sign of QE.

Q crosses zero when the scanning vector, P, comes into the plane containing the two vectors VL1 and VL2. Since we are interested only in the zero crossing, the magnitude of the scalar triple product may be scaled arbitrarily. This allows the VL and P vectors to be independently multiplied by non-zero scalars. In practice, the VL vectors are normalized and the P vector is expressed relative to the display dimension d.

The OGU's contain an edge generator for each of the 240 possible edges in the environment. The edge generator carries out the calculations required to determine the line and element numbers of the edge image on the display plane. Specifically it solves the equation:

$$J_0 = A + IB \tag{5}$$

where

$$A = \frac{QC}{QE}$$

$$B = \frac{QL}{QE}$$

Note that this is the equation of a line in a two-dimensional coordinate system whose axes are the top edge (zeroth line) and the left side (zeroth element) of the display plane. The quantity A is the intercept of the line with the J axis. B is the slope in elements per line.

When B is very large, $J_0$ is not of interest because it changes radically from line to line. This situation corresponds to an edge image which is nearly aligned with a raster line.

•We will compute $J_0$ only for those cases where the edge image crosses two or more raster lines; otherwise we will compute the line number which satisfied the equation:

$$I_o = -\frac{QC}{QL} \tag{6}$$

We thereby limit the range of the numbers which must be accommodated.

The edge generator, therefore, operates in two modes. In the first mode, it solves for an element number for each line, I. The second mode is employed when the edge image can be effectively represented by a single raster line. The VCU computes and transmits two quantities, A and B, to each edge generator. The A and B quantities are defined differently for the two modes of operation to simplify hardware implementation.

$$\text{Mode I} \begin{cases} A = \dfrac{QC}{QE} \\ B = \dfrac{QL}{QE} \end{cases} \tag{7}$$

$$\text{Mode II} \begin{cases} A = \dfrac{QC}{QL} \\ B = 1 \end{cases}$$

In Mode I, the B quantity is successively added to the A quantity each line time and the accumulated result for any line is the element number of the edge image. In Mode II, the A quantity is the negative of the raster line number. B is added to A as in Mode I until the latter changes sign on the line nearest the edge image.

Two indicator bits are appended to the B quantity for edge generator control purposes. The first is a mode indicator bit; a zero for Mode I and a one for Mode II. The second bit is a sign indicator. Recall from equation (4) and the conditions expressed by (1) that it is necessary to know the sign of Q at the start of each line. This sign cannot be determined from the A quantity alone. Therefore, in Mode I the sign of QE is put in the indicator bit and in Mode II the sign of QL is used. The edge generator compares the sign of the A register to the indicator bit to determine on which side of the edge a raster line starts and sets the initial state accordingly.

VCU arithmetic operations for edge generation may be summarized as follows. The vertices of all objects are stored for a particular environment. Each object is specified with respect to either the R coordinate system or one of the dynamic systems, and the vertex coordinates are V vectors in the respective systems. Since there are generally many V vectors, it is convenient to compute in the object coordinate systems since it is easier to transform three P basis vectors and one L vector to the object system than to convert all of the V vectors.

The L vector is first added to all V vectors in a particular coordinate system to give a complete set of VL vectors. The VL vectors are normalized. Pairs of VL vectors are then taken according to the object configuration and their cross products are computed. Each cross product is dotted with the three basic P vector components to form QC, QL and QE. The ratio of QL to QE is examined to determine the appropriate edge generator mode. The divisions indicated by equations (7) are performed and the A and B quantities are stored in an output table.

The order in which the VL vectors are used is a function of the composition of the environment. The VCU programs contain a separate subroutine for each type of solid and planar object which operates on the edges in a prescribed manner. The programmer need only be concerned with a calling sequence which specifies the order in which objects are used.

It was noted in the previous section that the outputs of the edge generators are combined logically (in the OGU's)to form faces. It is necessary to send control information to the face-forming logic so that it may test appropriately for either a front or back face condition. This is necessary because the display plane image of a front face which is behind the station point (and hence not visible) is a back face, as may be seen by projecting the vertices through the station point. The resulting ambiguity may be eliminated by first determining on which side of the face the station point lies.

This computation, termed "perspective", must be made for each face. A vector normal to each face is precomputed and stored as input data along with the vertices. The dot product is formed between the stored normal and a VL vector to one of the vertices on the face. The sign of the result specifies whether the front or back side is potentially visible.

### 2.1.2 Priority

Given a particular object type (tetrahedron, hexahedron, etc.) the results of the perspective computation for each face of the object, and the outputs of the edge generators, the OGU can construct the display plane image of the object. When additional objects are considered, it is necessary to know which objects obscure others. Priority need only be determined on an object basis (rather than on a face basis) since concave and intersecting objects are not permitted. Furthermore, objects can appear only in the view to which they are assigned, and priority considerations apply only between objects assigned to a particular view.

In general it would be necessary to check each object against a maximum of thirty-nine other objects to obtain complete priority relationships. Since the priority comparisons depend on the environment layout, they must be programmed specifically for each environment. The number of comparisons required varies as the square of the number of objects involved. Rather than consider priority in this manner, which would result in a lengthy priority program and unwieldy hardware interconnections, priority is handled in two steps. Object by object comparisons are made between objects within each OGU so that we need work only with two sets of twenty objects each. This about halves the number of comparisons required in the worst case. Priority between the two OGU's is called second-level blanking. A single determination specifies which OGU has priority over the other. It is necessary to construct the environment in such a way that this division between OGU's can always be made. A sufficient condition is that a convex envelope can be placed around the set of objects in each OGU and that these two envelopes are always non-intersecting.

In determining priority between objects, it is necessary to find planes which separate each object from its neighbors. Once these planes are specified for an environment, priority may be computed by calculating the position of the station point relative to the planes; for example, a front/back determination can be made. Object faces themselves can, in almost all instances, be found to serve as the separating planes. The results of the perspective computations may then be used for assigning priority. If an object face cannot be found to separate two objects, or if more efficient separation can be achieved, an auxiliary plane can be defined. This plane's normal would be stored along with object data. The priority program consists of a set of instructions which test the perspective words and form priority words from them.

Second level priority is computed in the same way except that two object groups are involved, each of which consists of all the objects in the OGU. If the objects are all in the same coordinate system (so that they cannot move relative to one another), then a fixed dividing plane can be found. If the OGU objects are in different coordinate systems, then the separating plane must be computed. A simple method is to assume that the plane is half way between the origins of the two object systems and perpendicular to the line connecting them.

## 2.2 R520 ARITHMETIC OPERATIONS

### 2.2.1 L Vectors

The R520 must compute sets of L and P vectors for subsequent use by the VCU and SGS. These quantities are derived from source computer inputs and stored initial condition quantities. The following discussion describes the chain of computations. Certain special purpose calculations, not directly involved in forming the L and P vectors, are described separately.

The program for the R520 is written for operation with up to three dynamic coordinate systems. One view is located in each dynamic system. With this configuration, one may construct a single environment with three interacting views, three entirely independent environments for separate studies, or various combinations. Figure 5 shows the coordinate system and view configuration and shows some typical vectors which relate them.

An L vector must be computed between each station point and the origin of every coordinate system which is to participate in that view. L vectors are transformed so that they are expressed relative to the coordinate system to which they point. The L vectors are then in coordinate systems compatible with the V vectors. The data blocks in the R520 are arranged to accommodate all possible L vectors for the three views and three dynamic systems - a total of nine. However, the programs are written under the assumption that objects are located in the first two dynamic systems and in the reference system. The third dynamic system is used only to move view C. No L vectors to this system are computed.

It can be seen from Figure 5 that the L vectors can be formed from the SP vectors, which are supplied as initial conditions, and the T vectors which depend upon these initial conditions and the source computer inputs. In the following equations, a vector quantity in brackets indicates a column vector and a parenthetical letter or number specifies the coordinate system of the quantity. An angle sequence of $\psi\,\theta\phi$ is assumed for purposes of this discussion.

The SP vectors are specified in the dynamic system of the view and locate the viewing point relative to the system origin. They are transformed to R coordinates:

$$\begin{aligned}(SPA(R)) &= (-\psi 1)(-\theta 1)(-\phi 1)(SPA(1))\\ (SPB(R)) &= (-\psi 2)(-\theta 2)(-\phi 2)(SPB(2))\\ (SPC(R)) &= (-\psi 3)(-\theta 3)(-\phi 3)(SPC(3))\end{aligned} \tag{8}$$

The SP and T vectors may now be combined to form L vectors in R coordinates. LA2, for example, is:

$$LA2\ (R) = T2 - T1 - SPA(R) \tag{9}$$

Other L vectors are similarly computed. The three L vectors to the R system are now in the proper coordinate system. The others require further transformation. Again using LA2 as an example:

$$LA2(2) = (\phi 2)(\theta 2)(\psi 2)\ LA2\ (R) \tag{10}$$

LB1, LC1, and LC2 undergo similar transformations--the first two into system 1 and the last into system 2.

### 2.2.2 Roll

Before the P vectors are computed the display plane roll angle must be found. Display roll is computed only if the view is to operate with the SGS; otherwise, the angle is set to zero. Operation with the SGS requires that the display raster be rotated so that the raster lines are parallel to the X-Y plane of the reference system. To simplify implementation of horizon curvature, we further specify that the roll angle be such that the raster is scanned from the sky (if it appears) to the ground. These conditions may be equivalently stated by saying that the display is rolled until the reference Z axis has no projection on the display v axis and its projection on the w axis is non-zero.

Consider the computation required for view A. A unit vector along the Z axis is first transformed into unrolled display coordinates:

$$\begin{pmatrix} ZOAU \\ ZOAV \\ ZOAW \end{pmatrix} = (\beta A)(\alpha A)(\phi 1)(\theta 1)(\psi 1)\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{11}$$

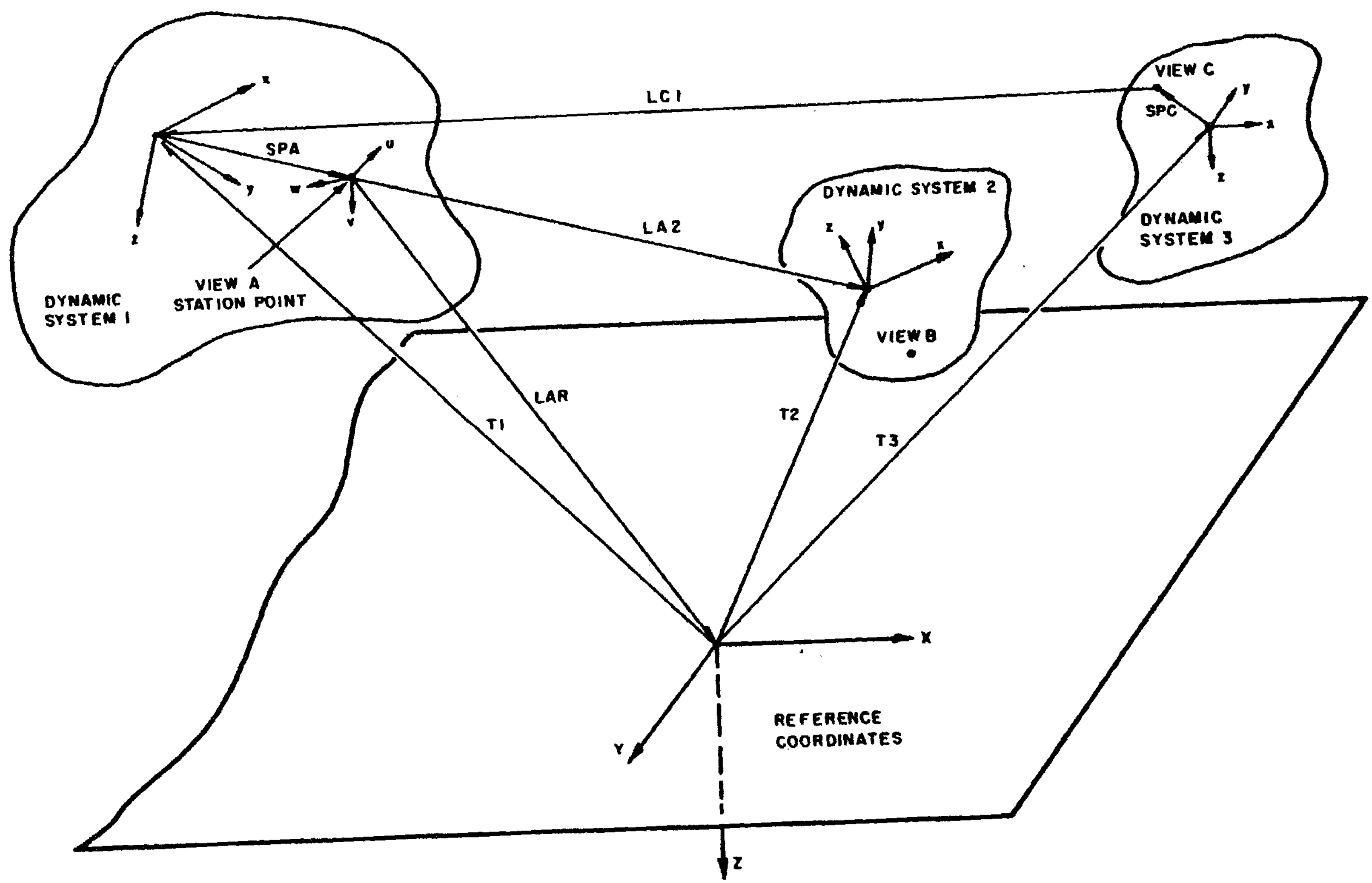


Figure 5. Coordinate Systems

The sine and cosine of the required roll angle are given by:

$$\mathrm{SIN}\,\gamma\;A = -\frac{\mathrm{ZOAV}}{\left|\left[(\mathrm{ZOAV})^2 + (\mathrm{ZOAW})^2\right]^{\frac{1}{2}}\right|} \tag{12}$$

$$\mathrm{COS}\,\gamma\;A = \frac{\mathrm{ZOAW}}{\left|\left[(\mathrm{ZOAV})^2 + (\mathrm{ZOAW})^2\right]^{\frac{1}{2}}\right|}$$

### 2.2.3 P Vectors

A set of basic P vector components, PC, PL, and PE, is stored for each view. These stored constants are determined by the number of lines and elements and by the angle which the display plane is to subtend. Each vector set must be transformed into the coordinate systems which can be seen by the view, just as were the L vectors.

Transformation of the P vectors involves the display angles $\alpha$, $\beta$, $\gamma$, and $\delta$. $\alpha$ and $\beta$ are sight line angles which are determined by the user for each problem. The display roll angle is computed to satisfy SGS requirements. The horizon dip angle is also a computed quantity and its derivation is discussed below. This is the final rotation of the display sequence -- a rotation about the v axis which effectively depresses the ground plane about an axis in the plane which passes through the nadir and is parallel to the v axis. The dip angle rotation is only used for P vectors in the reference system since this is the ground system.

The P vectors for view A must be expressed in the reference system and dynamic system 2 (there are no objects in system 2). The rotations for the vector PCA are:

$$\begin{aligned}(\mathrm{PCAR}) &= (-\psi 1)(-\theta 1)(-\phi 1)(-\alpha A)(-\beta A)(-\gamma\, A)(-\delta\, A)\,(\mathrm{PCA}) \\ (\mathrm{PCA2}) &= (\phi 2)(\theta 2)(\psi 2)(-\psi 1)(-\theta 1)(-\phi 1)(-\alpha A)(-\beta A)(-\gamma A)(\mathrm{PCA})\end{aligned} \tag{13}$$

PL and PE vectors undergo identical rotations. The other display P vectors are transformed in a similar manner for a total of seven P vector sets.

## 2.3 SGS ARITHMETIC QUANTITIES

The L and P vectors to the reference system are transmitted to the SGS for each view. These quantities correspond closely to those previously computed by the Program Control Unit of the SGS. In fact the bulk of the SGS computations now takes place in the R520. Part of the remaining computing capacity of the SGS is devoted to horizon curvature computation.

## 2.4 SPECIAL COMPUTATIONS

### 2.4.1 Beacon

The system has the capability of displaying one blinking beacon. The beacon may be placed in any of the coordinate systems and assigned to any view. The beacon is assumed to be a point source and its image size is fixed at one element by one line pair. It is generated by a special beacon generator card in an OGU and it is treated as another object for purposes of priority.

The computation consists of finding the element and line number for the beacon and providing an on/off control bit. The beacon program is written for a beacon in dynamic system two, viewed from system one. The computation is described in these terms to use a specific example. Modification of the program for other configurations is a matter of using the appropriate L vector and coordinate rotations.

The beacon position is specified by a vector to the beacon position from the origin of the coordinate system in which it is located. The components of this vector are IPCNX, Y, and Z and are entered as initial conditions. A vector BCN is formed from the viewing station point to the beacon:

$$BCN = LA2 + IBCN \tag{14}$$

The vector must be transformed from system two coordinates into View A display coordinates. The transformation is simplified by first normalizing the vector (its length being of no concern). The following rotations are then performed:

$$\begin{pmatrix} BCNU \\ BCNV \\ BCNW \end{pmatrix} = (\gamma A)(\beta A)(\alpha A)(\phi 1)(\theta 1)(\psi 1)(-\psi 2)(-\theta 2)(-\phi 2) \begin{pmatrix} BCNX \\ BCNY \\ BCNZ \end{pmatrix} \tag{15}$$

The beacon element and line numbers are:

$$\begin{aligned} J &= \frac{m}{2}\left(1 + \frac{P}{d}\ \frac{BCNV}{BCNU}\right) \\ I &= \frac{n}{2}\left(1 + \frac{P}{d}\ \frac{BCNW}{BCNU}\right) \end{aligned} \tag{16}$$

The beacon is displayed only if ICNU is positive, indicating that the beacon is in front of the viewer.

Blinking is controlled by two software counters, one for the ON time and one for the OFF time. The counters are modified each frame and the contents are compared against constants which determine the ON and OFF times in integral multiples of the frame time.

2.4.2 Dynamic Shadow

The shadow cast on the ground plane by a model of one's own vehicle is computed. It may be seen only by the view located in the dynamic system of the vehicle. To simplify the computations, the vehicle is modeled by a quadrilateral. The shadow, also a quadrilateral, is an accurate projection of the shadow - casting surface onto the X-Y plane of the reference system. It takes priority over the textured ground plane and all objects but will not be a proper projection when it falls on surfaces other than the ground plane.

The vertices of the quadrilateral model enter into the computation as initial conditions, IVS1 through IVS4. The size and shape may be specified to match the vehicle as desired. The X component of the vertices is normally made zero to keep the model parallel to the display plane. The sun azimuth (SUNA) and elevation (SUNE) angles must be specified within the limits: $0 \leq \text{SUNA} < 360^\circ$, $4^\circ \leq \text{SUNE} \leq 90^\circ$. The azimuth angle is measured from the X axis (north), $270^\circ$ indicating a western sun.

Both the R520 and the VCU enter into the computation. Rotation of the VS vectors into R coordinates and the computation of two trigonometric functions is done in the R520. The VCU forms the VL vectors to the shadow vertices. The shadow programs are written for view A. Figure 6 shows the geometry of the problem. Parallel sun rays project through the vertices to their shadow points. The shadow of vertex 1, for example, may be found as follows. IVS1 is first transformed to R coordinates:

$$\begin{pmatrix} VS1X \\ VS1Y \\ VS1Z \end{pmatrix}_{(R)} = (-\psi 2)(-\theta 2)(-\phi 2)(-\alpha A)\,(-\beta A) \begin{pmatrix} IVS1X \\ IVS1Y \\ IVS1Z \end{pmatrix}_{(D)} \tag{17}$$

The coordinates of the station point shadow relative to the station point are given by:

$$\begin{aligned} X &= LARZ \cdot CTN\,(SUNE)\,COS(SUNA-\pi) = LARZ\ CSC \\ Y &= LARZ \cdot CTN\,(SUNE)\,SIN\,(SUNA-\pi) = LARZ\ SSC \\ Z &= LARZ \end{aligned} \tag{18}$$

where

$$CSC = -CTN\,(SUNE)\,COS\,(SUNA)$$
$$SSC = -\,CTN\,(SUNE)SIN\,(SUNA)$$

The VL vector to vertex 1 of the shadow is then found by adding to the X and Y coordinates of the station point shadow, 1) the components of vector VS1(R), and 2) the additional components obtained by casting a shadow of the tip of the VS1 vector displaced to the station point shadow. It is:

$$\begin{aligned} VL1X &= [LARZ - VS1Z]\,CSC + VS1X \\ VL1Y &= [LARZ - VS1Z]\,SSC + VS1Y \\ VL1Z &= LARZ \end{aligned} \tag{19}$$

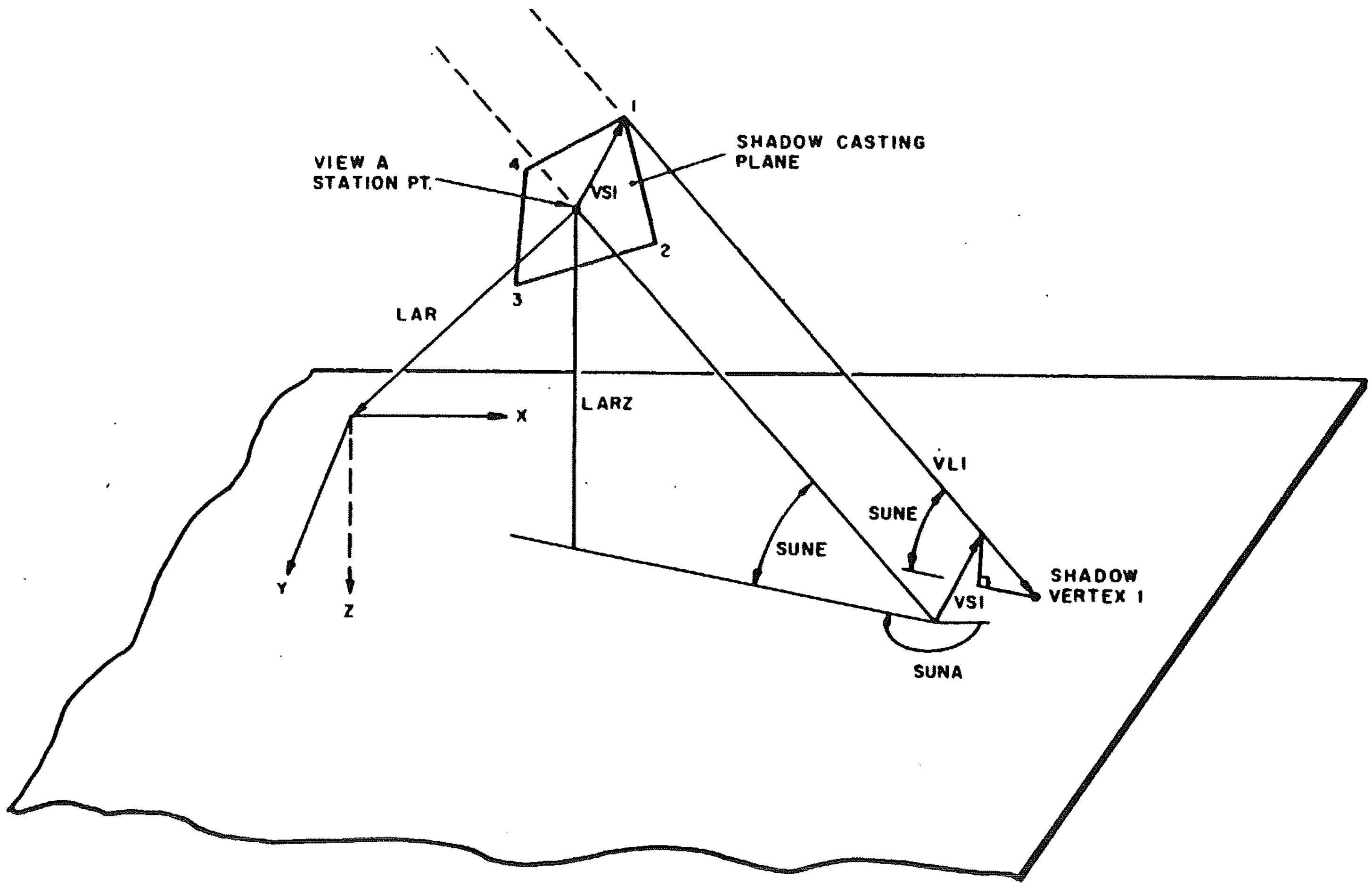


Figure 6. Shadow Geometry

The other vertices are computed in an identical manner. The rotations of equation (17) are performed by subroutine RRN in the R520. The two constants, SSC and CSC, are formed by subroutine SUN. The VL vectors are computed in the VCU from equations (19).

### 2.4.3 Horizon Dip and Curvature

In some simulations, operation with a flat ground plane is a serious handicap -- particularly when the mission requires that sightings be taken on the horizon. This problem is overcome by causing the horizon to have the curvature and depression angle that would be observed if the ground were modeled by a sphere.

Figure 7 illustrates the method of curving the horizon line. Because of the symmetry involved, it is convenient for this discussion to use the i-j coordinates shown, which are centered on the display plane. Since the SGS is in use, the display coordinates have been rolled so that the raster lines are parallel to the normal horizon. Curvature is introduced by blanking out sections of the ground plane as shown. Since these sections are symmetrical about the i axis, it is necessary to compute one j quantity for each line below the horizon up to the point where the horizon goes off of the display.

Depression of the horizon is accomplished by the method shown in Figure 8. In this cross-sectional view an observer is at an altitude h above the ground and looking straight ahead. The horizon line of the infinite plane surface appears in the center of his view. The horizon of the spherical body is depressed by an angle $\delta$ . The same effect may be achieved by rotating the plane surface about an axis perpendicular to the page and passing through the nadir until it is parallel to the line going to the actual horizon.

An equation describing the image of the horizon of the spherical body in terms of i and j is given by:

$$j^2 = i^2 \left(\frac{m}{n}\right)^2 \left(\frac{r}{2h} \cos^2 \xi - \sin^2 \xi\right) - i \frac{m^2 pr}{nd^2 h} \sin \xi \cos \xi$$

$$+ \frac{p^2 m^2}{d^2 n} \left( \frac{r}{2h} \sin^2 \xi - \cos^2 \xi\right) \qquad (20)$$

where $\xi$ is the relative pitch angle which is measured between the sight line and the plane tangent to the sphere at the nadir. This is an exact solution for the curve.

The horizon depression angle may be obtained by solving for the angle $\xi$ which places the horizon in the center of the screen. Setting the last term in equation (20) to zero and solving for the angle:

$$\xi \Big|_{i = j = o} = \tan^{-1} \left(\frac{2h}{r}\right)^{\frac{1}{2}} \qquad (21)$$

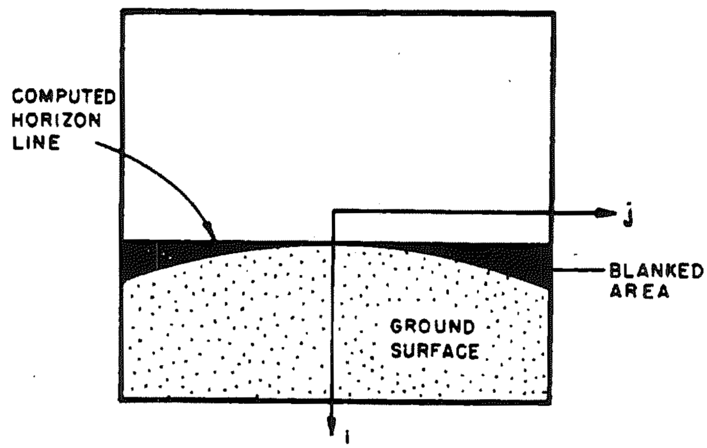


Figure 7. Horizon Curvature

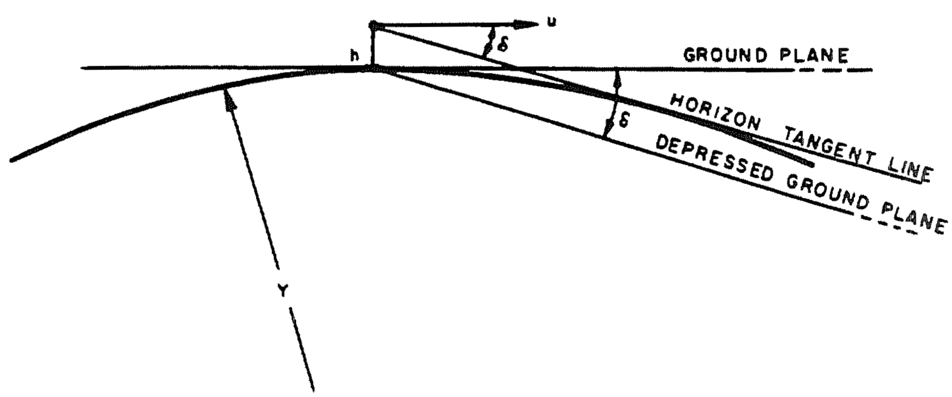


Figure 8. Horizon Depression

At this point some limits must be placed on the quantities of equation (20) to obtain an acceptable approximation to the curvature. Since operation is with the SGS, the largest altitude is $2^{17}$ feet. The radius of the spherical body will range between that of the moon and that of the earth. The ratio p/d is determined by the 25- and 60-degree fields of view to be computed. With a maximum field of view of 60 degrees, the horizon image shape is essentially independent of the angle $\xi$; the curve merely moves up and down the display plane. Due to this simplification, equation (20) need be solved only for the case where the curve is in the center of the display:

$$j^2 = i^2 \left(\frac{m}{n}\right)^2 \left(\frac{r}{2h} - 1\right) + i \, \frac{m^2 p}{nd} \sqrt{\frac{r}{2h}} \tag{22}$$

The curve is, therefore, a function only of altitude and of the display size. Equation (22) may be rewritten to make the curve independent of horizon starting point:

$$j_n = \pm \left[K_1 \, i_n^2 + K_2 \, i_n\right]^{\frac{1}{2}} \tag{23}$$

where the two constants are functions of altitude and the display size. $j_n$ is the element number from the center of the display for the $n^{th}$ line below the horizon.

It can be shown that, within the constraints mentioned above, the ratio $j_n/j_1$ for all values of $j_n$ which appear on the display is essentially independent of altitude. Therefore, if the ratios are computed for all n at the maximum altitude and these constant ratios $c_n$ are stored, the curve may be found for any altitude by computing $j_1$ from equation (23) and deriving the others from:

$$j_n = c_n \, j_1 \tag{24}$$

The error involved in the above approximations is less than one-half of the display line spacing.

Equation (23) is solved for $j_1$ in the R520. The constants $c_n$ are stored in the SGS and the multiplications implied by (24) are performed there. Two sets of $j_n$ numbers are computed for each view: one for the even field and one for the odd field. In order to determine which set goes with which field, it is necessary to compute the field on which the horizon line starts. This determination is made in the R520 by simulating the SGS processing of the quantities PCZ and PLZ.

The dip angle is computed in the R520 starting with equation (21). This is the angle which will depress the horizon properly when it is in the center of the display. However, the whole plane is at an angle. In order to avoid peculiar effects when the sight line is pointed down toward the plane, it is desirable to gradually reduce the dip angle to zero as the sight line approaches the nadir. A suitable function for this purpose is already available from the roll angle computations. The dip angle is therefore computed from:

$$\delta = (1 - ZOU)^2 \tan^{-1} \left(\frac{2h}{r}\right)^{\frac{1}{2}} \qquad (25)$$

The first two terms in the expansion of the inverse tangent function suffice for the range of angles encountered.

## 3.0 SAMPLE ENVIRONMENTS

Figures 9 through 14 in the pictorial appendix (pages A-1 through A-4) illustrate representative environments which can be produced by the ESG.

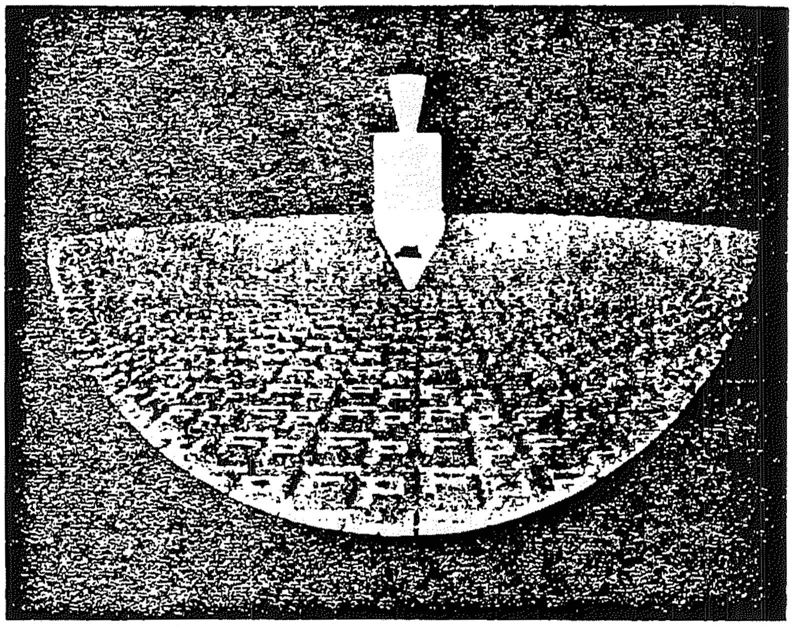

Figure 9. Command and Service Module
(240 edges)

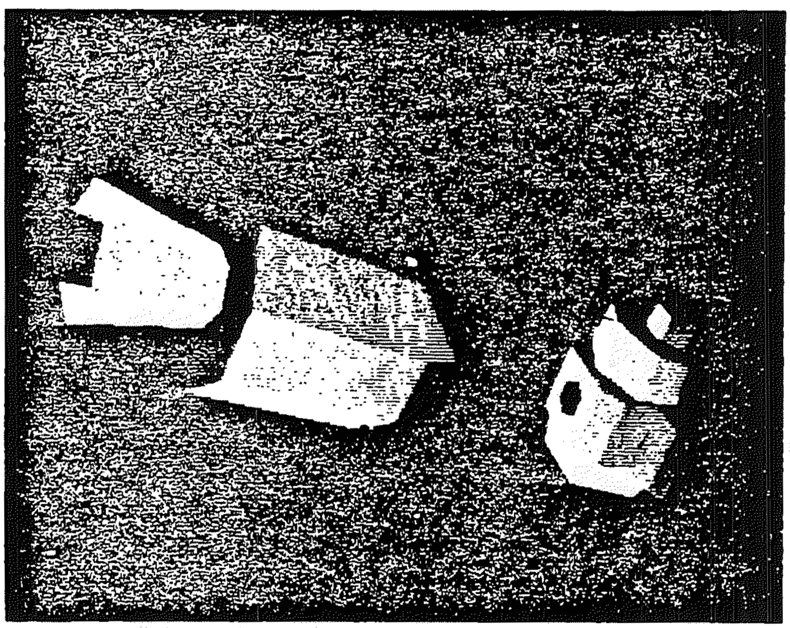

Figure 10. LM - CSM, Single View

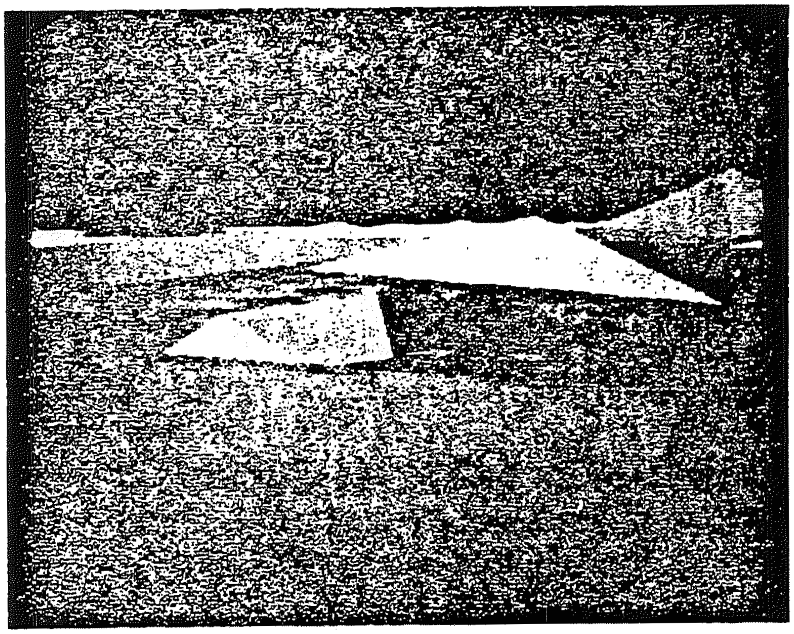

Figure 11. Lunar Environment with Crater

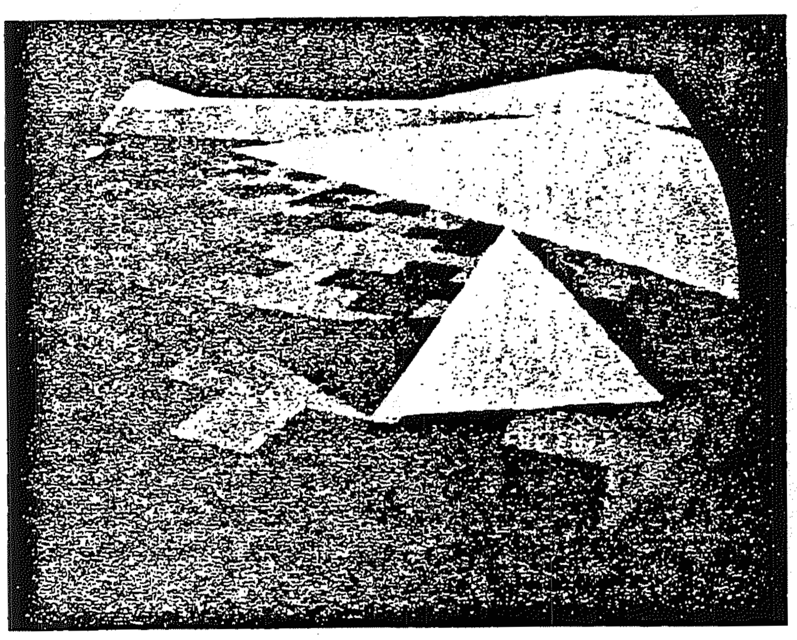

Figure 12. Lunar Environment with Object Shadow

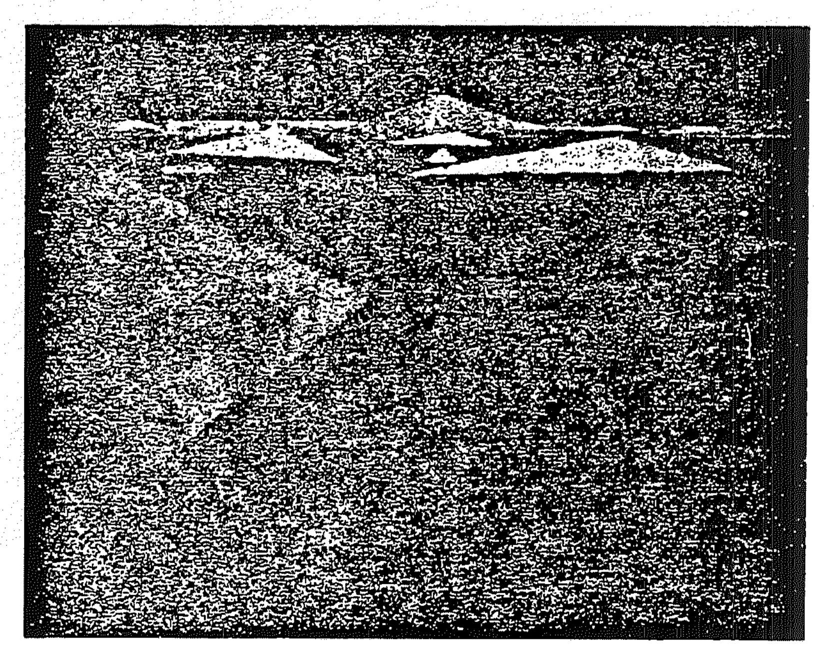

Figure 13. Lunar Environment with Own Ship's Shadow

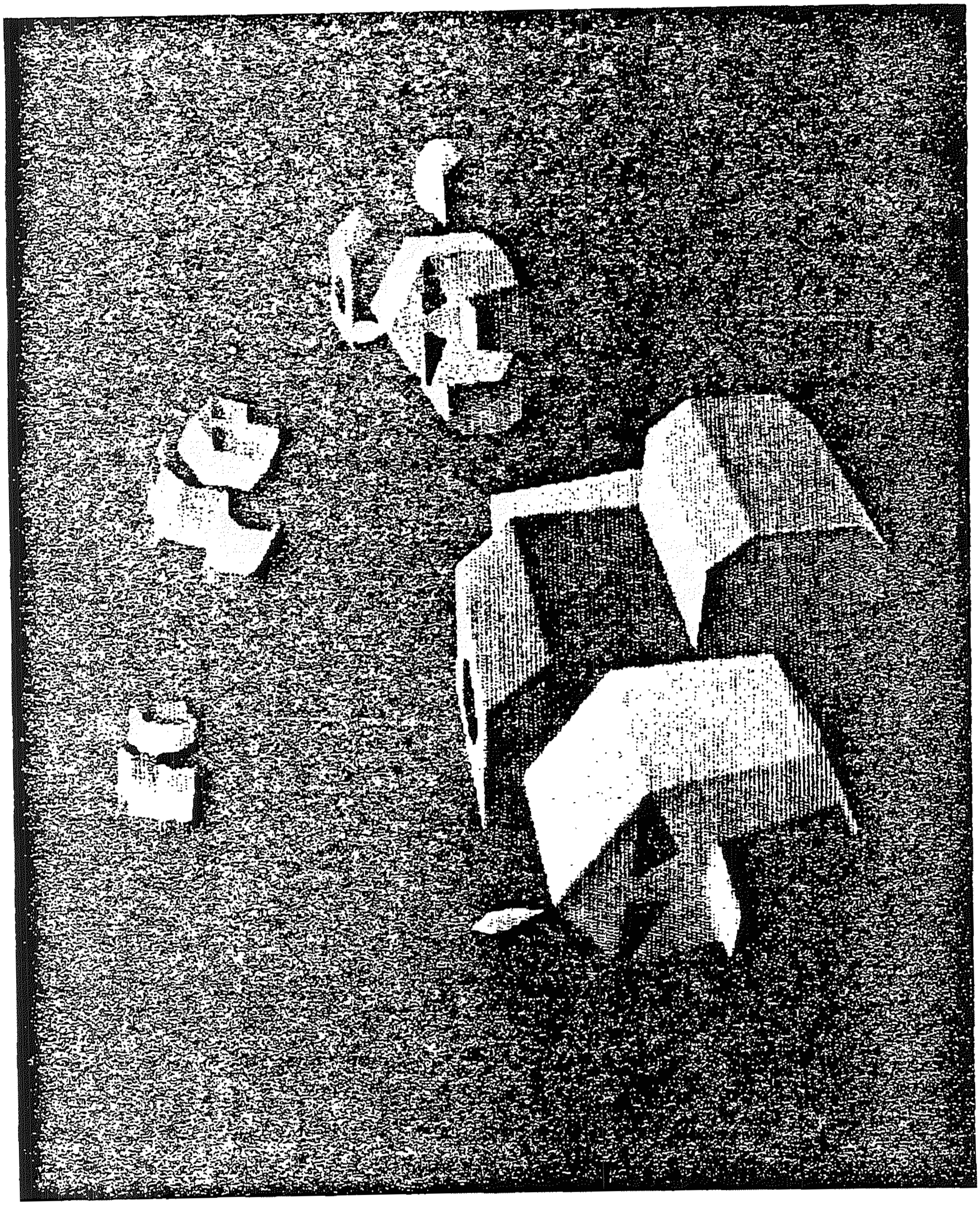

Figure 14. Lunar Module, Sequential Exposure

*N7173772*

*BX*

| | | |
|---|---|---|
| BIN: | M356 | 10-07-98 |
| INVOICE: | 606190 | |
| SHIPTO: | 1*85776 | |
| PAYMENT: | NONE | |